Assignment 2 - Hibernate

Submission instructions

The assignment should be submitted by pairs. You should submit project jar file (Java) no need to submit tables and triggers and jar files.
The jar file should contain your project: all Java/XML sources, without external JARs.
The name of your project and jar file should be of form id1_id2 where id is your ID number.
Dead line: 24/12/2017

1. Development instructions

   All DB functions should use Hibernate and HQL.

   Make sure you have the table from the previous assignment (MediaItems, Similarity).

   1.1. Oracle SQL Developer

   1.1.1. Create a table Users with the following columns:
   - USERID (NUMBER(9,0)) –primary key
   - USERNAME (VARCHAR2(200))
   - PASSWORD (VARCHAR2 (200))
   - FIRST_NAME(VARCHAR2 (200))
   - LAST_NAME(VARCHAR2 (200))
   - DATE_OF_BIRTH (TIMESTAMP)
   - REGISTRATION_DATE(TIMESTAMP)

   1.1.2. Create a table Administrators with the following columns:
   - ADMINID (NUMBER(9,0)) –primary key
   - USERNAME (VARCHAR2(200))
   - PASSWORD (VARCHAR2 (200))

   1.1.3. Create a table LoginLog with the following columns:
   - USERID (NUMBER(9,0)) –primary key (foreign key to Users)
   - LOGINTIME (TIMESTAMP)-primary key

   1.1.4. Create a table History with the following columns:
   - USERID (NUMBER(9,0)) –primary key (foreign key to Users)
   - MID (NUMBER(9,0))-primary key (foreign key to MediaItems)
   - VIEWTIME (TIMESTAMP)-primary key

1.2. Java

1.2.1. Add to your project Hibernate jars. Generate using Hibernate tool the ORM.

1.2.2. Make new Java class Assignment.java

1.2.3. Write the Java function isExistUsername

```
public static boolean isExistUsername (String username)
```

- The function returns true if the received username exist in the table USERS otherwise false.

1.2.4. Write a Java function insertUser

```
public static String insertUser(String username, String password, String
first_name, String last_name, String day_of_birth, String
month_of_birth, String year_of_birth)
```

- The function checks if the username exist in the USERS table, in case of positive answer returns null else insert the user to the table USERS including registration_date field and returns the user id (USERID).
- Use generator class="increment" for USERID.

1.2.5. Write a Java function getTopNItems

```
public static List<Mediaitems> getTopNItems (int top_n)
```

- The function retrieves from the table MediaItems first top_n items (mid ascending order).

1.2.6. Write the Java function validateUser

```
public static String validateUser (String username, String password)
```

- The function compares received values with existing in the data base.
- The function return USERID if the values are equal to the values in the table otherwise NULL.

1.2.7. Write the Java function validateAdministrator

```
public static String validateAdministrator (String username, String
password)
```

- The function compares received values with existing in the data base.
- The function return ADMINID if the values are equal to the values in the table otherwise NULL.

1.2.8. Write the Java function insertToHistory

```
public static void insertToHistory (String userid, String mid)
```

- The function insert the row to the History table with current server time.

1.2.9. Write the Java function getHistory

```
public static List<?> getHistory (String userid)
```

- The function retrieves from the tables History and MediaItems users's items.
- The function return List of pairs <title,viewtime> sorted by VIEWTIME in descending order.

1.2.10. Write the Java function insertToLog

```
public static void insertToLog (String userid)
```

- The function insert the row to the LoginLog table with current server time.

1.2.11. Write a Java function getNumberOfRegistredUsers

```
public static int getNumberOfRegistredUsers(int n)
```

- The function receives an integer number n
- The function retrieves from the table Users number of registered users in the past n days
- The function return integer number

1.2.12. Write a Java function getUsers

```
public static List<Users> getUsers ()
```

- The function retrieves from the table Users all users
- The function return List of objects Users

1.2.13. Write a Java function getUser

```
public static Users getUser (String userid)
```

- The function retrieves from the table Users user's information
- The function return objects Users

Good Luck,
Ben Lerer & Dana Behnam.